

Structure identification methods for atomistic simulations of crystalline materials

Alexander Stukowski

Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

E-mail: alexander@stukowski.de

Abstract. We describe a new computational analysis technique to identify lattices and structural defects in large-scale atomistic computer simulations of crystalline materials. Our approach is based on a user-supplied catalog of template structures. We use a graph-based pattern matching algorithm to find occurrences of periodic and non-periodic atomic arrangements in atomistic snapshots, and to generate a high-level description of a simulated microstructure. The method covers defects such as stacking faults, grain boundaries, crystal interfaces, point defects, and defect clusters in a wide range of crystal lattices. In contrast to existing methods, the proposed pattern matching algorithm is able to identify crystal structures with a polyatomic basis. Furthermore, we discuss how the local lattice orientation can be determined to measure crystal rotations, and how a defective crystal can be mapped to an ideal reference state. Finally, we derive a computational method for detecting and characterizing disclination defects in the analysis data.

Submitted to: *Modelling and Simulation in Materials Science and Engineering*

1. Introduction

Atomistic simulation methods such as molecular dynamics (MD), molecular statics, and Monte Carlo schemes are routinely used to study crystalline materials at the atomic scale. As crystal defects play a critical role for the understanding of most materials properties, they have been the subject of a large number of simulation studies. In this paper we present a set of new computational analysis techniques that allow one to extract and characterize structural defects in large-scale atomistic computer simulations of crystalline materials in a fully automated fashion. The described methods cover defects such as stacking faults, grain boundaries, coherent crystal interfaces, point defects, and defect clusters in a wide range of crystal lattices.

In recent years, the availability of improved simulation methods, better interatomic potentials, and increased computing power has led to the expectation that atomic-scale computer simulations should yield exact quantitative output in addition to qualitative insights into physical processes. Obtaining quantitative data from, for instance, a molecular dynamics simulation usually requires processing the raw atomic trajectories in a specific way, thereby extracting information on the number, evolution, and interaction of important features such as crystal defects. Such data can finally be

used to interpret experimental measurements or, perhaps more importantly, provide input to coarse-grained models of materials behavior.

Existing analysis techniques such as the common neighbor analysis (CNA) [1] and the centrosymmetry parameter (CSP) [2] are conventionally used in simulation studies of crystal plasticity to filter out atoms which form a perfect lattice, thereby revealing crystal defects such as dislocations and grain boundaries. Even though this is usually sufficient for the scientist to visualize and visually interpret the important physical processes [3], it does not yield an abstract description of the system: The purely atomistic representation of the crystal is maintained by these analysis methods, and each atom is processed and classified independently. In particular, no higher-level (that is more abstract) description of the relevant crystal features (i.e. grains and lattice defects) constituted by the atoms is obtained through these simple techniques. Therefore, novel data-reduction approaches need to be developed, which are able to identify and characterize extended crystal defects efficiently and accurately, and transform them into discrete objects, which are accessible to quantitative analyses. This is needed to catch up with the increasing complexity of large-scale MD simulations, which model the interaction of a large variety and a large number of such crystal defects.

In this paper we describe new computational techniques for the fully-automated identification of grains with arbitrary lattice structures, coherent crystal interfaces including grain boundaries, planar stacking faults, and zero-dimensional crystal defects in atomistic simulation data. They are complemented by an already available analysis method for dislocations [4, 5]. In addition to identifying and classifying defects in the atomistic input data, the goals of the techniques developed in this paper are to determine the lattice orientation and shape of individual grains in a polycrystal, to determine the orientation of defects with respect to the parent lattice, and to measure elastic strains in the material. Another application described in this paper is the detection of disclinations in the crystal.

The data extracted and generated from an atomistic snapshot should be as comprehensive as possible to support unforeseeable types of queries conceived by the users. The general aim of the present effort is to reach a level of representation that is more abstract than the overloaded, fully atomistic description of the microstructure, but which retains all its important features such that, in principle, the atomistic system could be reconstructed from it (lossless data reduction).

A second design goal for an analysis code is universality. That is, the set of recognized lattices and defects is not hard-coded into the algorithm, but is extensible by the user, who may add new lattice types, interfaces and other defects to the search catalog simply by providing a template of the structure. From this template the algorithm automatically generates a characteristic pattern that is used to find occurrences of the structure in the actual simulation data.

We give an overview of the most commonly used methods for extracting structural features from atomistic simulation snapshots of crystalline materials in the following sections. We then discuss their shortcomings and limitations in section 1.2, which motivated the development of the new techniques presented in sections 2 and 3.

1.1. Existing analysis techniques

Here, we focus on analysis techniques for crystalline materials only. A broader overview of structural characterization methods for general particle systems can be found in [6].

1.1.1. Energy filtering The potential energy of an atom can be used as a simple indicator to decide whether it forms a perfect lattice with its neighbors. Given that crystal defects are usually higher in energy than the perfect lattice, one can detect the former by applying a threshold criterion to the atomic energies: Atoms having a potential energy above the threshold value are considered defect atoms, while low-energy atoms are classified as regular crystalline atoms.

Several shortcomings of this simple method have contributed to the fact that it is rarely used nowadays. Since both the perfect lattice state and the defect state of the crystal are usually local energy minima, the perfect lattice can remain stable at atomic energies close to or even above the defect energy. In particular, the discrimination becomes unreliable if the energy ranges of lattice and defect atoms overlap due to the effects of elastic strain energy or thermal energy.

Moreover, the potential energy of individual atoms is specific to the employed interaction model, and, for some interatomic potentials and quantum mechanical descriptions, is not computable at all. This is why one usually prefers structural analysis methods, which compare the spatial arrangement of atoms to a reference configuration.

1.1.2. Centrosymmetry parameter The centrosymmetry property of some lattices such as face-centered cubic (fcc) and body-centered cubic (bcc) can be used to distinguish them from other structures such as crystal defects where the symmetry is broken. Kelchner et al. [2] have developed a metric, the so-called centrosymmetry parameter (CSP), that quantifies the local loss of centrosymmetry at an atomic site, which is characteristic for most crystal defects.

The CSP of an atom having N nearest neighbors is defined as

$$CSP = \sum_{i=1}^{N/2} |\mathbf{r}_i + \mathbf{r}_{i+N/2}|^2 \quad (1)$$

where \mathbf{r}_i and $\mathbf{r}_{i+N/2}$ are vectors from the central atom to a pair of opposite neighbors. Practical ways of finding these pairs are described in the accompanying documentation of the visualization program ATOMEYE [7] and the molecular dynamics code LAMMPS [8].

The main advantage of the CSP is that it is affected only marginally by elastic distortions of the crystal. In particular, any affine deformation of the lattice does not change its degree of centrosymmetry at all. The CSP is, however, sensitive to random thermal displacements of atoms. Being only a one-dimensional measure, the CSP's ability to differentiate between different defect structures is rather weak. The noise induced by thermal displacements and inhomogeneous elastic strain may well dominate any characteristic differences between structures. Naturally, only centrosymmetric crystal lattices can be properly treated with this method, and it provides no means of differentiating between several fully centrosymmetric crystal phases.

1.1.3. Common neighbor analysis Analysis methods that employ more complex, multi-dimensional signatures to characterize arrangements of atoms are usually better in discriminating between several structures. A popular method of this kind is the common neighbor analysis (CNA) [1]. Particular to the CNA is that the characteristic signature is not a direct function of the atomic coordinates. Instead the CNA is based

on an abstract representation of the crystal's topology, the network of bonds between atoms.

Usually, two atoms are said to be near-neighbors (i.e. bonded) if they are within a specified cutoff distance of each other. For densely packed structures the cutoff distance lies halfway between the first and second neighbor shell of the lattice under consideration, while for the bcc lattice (and other more open lattices) several shells need to be taken into account.

To assign a local crystal structure to an atom, three characteristic numbers are computed for each of the N neighbor bonds of the central atom: The number of common neighbors, the total number of bonds between these common neighbors, and the number of bonds in the longest chain of bonds between the common neighbors. This gives N triplets (ijk) , which are compared to the reference patterns of typical lattice structures (table 1):

fcc ($N = 12$)	hcp ($N = 12$)	bcc ($N = 14$)	diamond ($N = 16$)
$12 \times (421)$	$6 \times (421)$	$8 \times (666)$	$12 \times (543)$
	$6 \times (422)$	$6 \times (444)$	$4 \times (663)$

Table 1. CNA signatures for common crystal structures.

The *common neighborhood parameter* [9] should be mentioned as an alternative analysis method, which was proposed by Tsuzuki et al. to combine the strengths of both the CNA and CSP methods. The CNA has also been extended to binary atomic systems by taking the chemical species of common neighbors into account as an additional criterion [10]. This extension enables the identification of simple binary structures such as $L1_0$, $L1_2$ etc.

1.1.4. Bond-angle distribution analysis The bond-angle distribution analysis has been developed by Ackland and Jones [11] to distinguish fcc, hcp and bcc coordination structures in an atomistic simulation. To this end, the $N(N-1)/2$ bond angle cosines $\cos \theta_{ijk}$ of an atom are used to build a histogram, which is then further evaluated using a set of heuristic decision rules to determine the most likely structure type. These criteria have been optimized such to archive a robust identification of the most important crystal structures.

1.1.5. Voronoi analysis The Voronoi decomposition has been employed in simulations of liquids and glasses to study various properties of their atomic structure [12, 13]. To characterize the local structural topology around a single particle, the corresponding Voronoi polyhedron is translated into a compact signature by counting the number of polygonal facets having three, four, five and six vertices. This yields a vector of four integers, (n_3, n_4, n_5, n_6) , that identifies the structural type. For instance the Voronoi polyhedron of an fcc lattice atom comprises 12 facets having four vertices each. Thus the corresponding signature is (0,12,0,0). The polyhedron of a bcc atom has facets with four and six vertices. The corresponding signature is (0,6,0,8).

Even though the Voronoi method has been used numerous times for the analysis of unstructured particle systems such as liquids and glasses, it has rarely been applied to simulations of crystalline materials. The highly symmetric crystalline packings in lattices such as fcc and hcp can cause singularities in the Voronoi construction, and

only minor perturbations in the atomic coordinates dramatically change the Voronoi polyhedra [14, 15]. This sensitivity and the high computational cost of the Voronoi polyhedron construction render the application of this method to crystalline systems less attractive.

1.2. Limitations of existing methods

The structure identification methods described above are restricted with respect to the types of crystal structures that can be identified:

- (i) The CSP method requires the local environment to be symmetric, and for the CNA neighbors need to be arranged on discrete shells around the central atom. Note that neither is generally the case for atoms in crystal defects.
- (ii) The bond-angle distribution method makes use of hard-coded rules to differentiate between lattice structures. These optimized rules were obtained heuristically for each supported lattice structure to yield optimal results. From a user's perspective, however, this manual approach severely limits the range of structures that can be treated with this method.
- (iii) The Voronoi method provides no means of consistently controlling the sensitivity of the structure identification, as even slightly distorted atomic positions tend to change the resulting polyhedra dramatically.

In addition, the existing crystal structure identification methods have two important limitations in common, which will be addressed by the present work:

- (i) The described methods are limited to simple lattice structures with a monatomic basis. Even though they can characterize the local coordination structure at each lattice site separately, they fail to identify lattices with multiple atoms per primitive unit cell and crystal defect structures consisting of several atoms.
- (ii) The described methods determine only whether an arrangement of atoms in a simulation snapshot matches a certain reference structure or not. To this end, the local arrangement of neighbors is transformed into a characteristic signature and compared to a known reference value, yielding either a match or a mismatch. While this is sufficient for simple visualization purposes, it is insufficient for comprehensive analyses as we pursue them. For many applications it is instrumental to associate the individual neighbor atoms with the corresponding bonds in the reference structure, i.e. to create a one-to-one mapping between the actual crystal and the ideal reference lattice. This extended information will enable us, for instance, to track the local crystal orientation, to detect dislocations and disclinations, and to calculate the elastic strain field.

2. Coordination pattern analysis

2.1. Overview

We address the aforementioned issues by proposing three computer algorithms, which, when put together, will provide a comprehensive analysis tool for simulation snapshots of crystalline materials.

We first describe the so-called *neighbor distance analysis* (NDA), which has been devised as an alternative to the conventional structure identification methods discussed in the review. While being computationally more expensive, it does not require the

arrangement of neighbor atoms to be symmetric or ordered in a specific way. The NDA provides a direct way of controlling the sensitivity of the matching process to thermal displacements and elastic deformations. It is most useful in situations where the coordination structure of atoms is lacking any particular symmetries or shell structure as it is usually the case in crystal defect cores.

Like other methods discussed above, the NDA operates on individual atoms only. In a second processing step, we will apply a pattern recognition method to identify extended structures such as complex lattices and grain boundaries that comprise multiple atoms. The task of this graph-based pattern search technique, described in section 3.3, is to find occurrences of periodic and non-periodic template structures in the simulation data. As a result of this second processing step, the input crystal is divided into clusters. Each cluster is a connected set of atoms that form a particular structure from the reference catalog (e.g. a certain lattice, grain boundary, stacking fault etc.) with long-range order.

In the last step, the orientational relationship between adjacent clusters is determined (section 4), and an abstract description of the entire microstructure is assembled. To give an example: If a grain boundary has been identified by the pattern recognition algorithm based on its characteristic atomic structure (for instance a twin boundary), we can infer the exact crystallographic orientation relationship between the two adjacent grains, irrespective of any elastic deformation that might be superimposed on the crystal. Ultimately, a high-level description of the polycrystal is obtained that no longer consists of individual atoms, but of discrete objects such as crystal grains, interfaces, and lattice defects.

2.2. Formal definition of a coordination pattern

A coordination pattern $p = (R_p, S_p)$ describes the local arrangement of neighbor atoms around a central atom in an arbitrary Cartesian coordinate system. The pattern is specified in terms of a list of bond vectors $R_p = (\mathbf{R}_1, \dots, \mathbf{R}_N)$ connecting the central atom with its N nearest neighbors (with N being a freely selectable parameter). The coordination pattern for fcc lattice sites, for instance, consists of $N = 12$ neighbors, with the neighbor vectors comprising the $a/2 \langle 110 \rangle$ vector family. Note that the bond list R_p has an arbitrary but fixed ordering.

The permutation group $S_p = \{\sigma_s\}$ describes the point symmetry of the coordination pattern, with $\{\sigma_s\}$ being a set of permutations of the numbers $\{1, \dots, N\}$, corresponding to permutations of the neighbors. For instance, the twelve neighbors of an fcc atom can be permuted in 48 equivalent ways, corresponding to the elements of the $m\bar{3}m$ point symmetry group of cubic crystals. The set S_p can be precomputed on the basis of R_p .

2.3. Coordination pattern matching

Let $V_a = \{a_i\}$ be the set of atoms in a simulation snapshot to be analyzed. Given an atom $a \in V_a$ and a coordination pattern p , the task of a *coordination pattern matching algorithm* is to determine whether the neighbor atoms of a form a structure that is sufficiently similar to the ideal arrangement p . The input arrangement to be tested against the pattern is given as a set of vectors $(\mathbf{r}_1, \dots, \mathbf{r}_N)$ connecting the central atom a with its N nearest neighbors $(a_1, \dots, a_N) \subseteq V_a$. The nearest neighbors of an atom in a simulation snapshot can be found by means of a k - d tree data structure [16] and

a recursive k -th nearest neighbor query algorithm [17], or by other means such as the neighbor lists in a molecular dynamics simulation.

In section 1.1 we have described various structure matching methods that all exploit structural symmetries in some way. Instead of directly comparing $(\mathbf{r}_1, \dots, \mathbf{r}_N)$ to $(\mathbf{R}_1, \dots, \mathbf{R}_N)$, they condense both sets into characteristic signatures which are invariant under rotation, and which can easily be compared. This transformation is essentially what makes the identification process efficient and robust (see [6] for an in-depth discussion). Note that, at the same time, this data reduction usually results in some insensitivity to elastic deformation: Small perturbations of the atomic positions do not change the calculated signature.

2.4. Neighbor distance analysis

In general, the atomic arrangements found in the core regions of crystal defects may not exhibit any symmetry or other type of order such as discrete neighbor shells. It might therefore be difficult to reduce their description to a small set of characteristic numbers (and even less so to a scalar signature like the CSP). Thus, in such a case, one has to go back to a more extensive type of signature, as we will propose it in the following.

For this, let us suppose the vectors of the coordination pattern, $R_p = (\mathbf{R}_1, \dots, \mathbf{R}_N)$, are ordered according to their distance from the central atom such that $R_1 \leq \dots \leq R_N$. Correspondingly, we can sort the (initially random) list of neighbor atoms too such that $r_1 \leq \dots \leq r_N$. Note that this is usually not sufficient for correctly associating the actual atoms with the reference vectors: The bond lengths may be perturbed by thermal displacements, and the ordering can be non-unique if neighbors are arranged on discrete shells. However, we may compute a local hydrostatic scaling factor, λ , from the two sorted bond lists:

$$\lambda = \frac{1}{N} \sum_{i=1}^N (R_i/r_i) \quad (2)$$

This scaling factor relates the lattice constant of the reference structure (which may be arbitrary, and is often chosen to be unity) to that of the actual crystal, which depends on factors such hydrostatic stress, temperature, and composition.

The mapping between the reference vectors $(\mathbf{R}_1, \dots, \mathbf{R}_N)$ and the neighbor atoms (a_1, \dots, a_N) , as we want to determine it, can be expressed in terms of a permutation $\sigma = (a_{j_1}, \dots, a_{j_N})$ of the original, randomly-ordered neighbor list. As discussed in section 2.2, multiple equivalent permutations may exist due to the symmetries of the coordination structure. Given any valid permutation map σ , we can obtain all other equivalent mappings by applying the precomputed symmetry permutations S_p to it.

How is a valid mapping σ determined? For this we define a new type of signature that is based on the linear distance $|\mathbf{R}_i - \mathbf{R}_j|$ between two neighbors i and j , which is invariant under rotation. Hence, we give this approach the name *neighbor distance analysis* (NDA). We assume that the coordination pattern ρ describes the equilibrium positions of atoms. In the actual crystal, however, atoms may be displaced due to thermal vibrations or elastic distortions of the lattice. Let the maximum allowed deviation of an atom from its equilibrium position be given by a user-definable parameter δ_{\max} . Then the test structure $(\mathbf{r}_1, \dots, \mathbf{r}_N)$ matches the reference pattern if

at least one mapping σ exists such that the condition

$$\underbrace{|\mathbf{R}_i - \mathbf{R}_j| - \delta_{\max}}_{d_{ij}^{\min}} \leq \underbrace{\lambda |\mathbf{r}_{\sigma(i)} - \mathbf{r}_{\sigma(j)}|}_{d_{ij}} \leq \underbrace{|\mathbf{R}_i - \mathbf{R}_j| + \delta_{\max}}_{d_{ij}^{\max}} \quad (3)$$

is fulfilled for all $N(N-1)/2$ neighbor pairs. That is, all rescaled distances must lie in the corresponding intervals given by the reference structure. This condition is illustrated by figure 1(a).

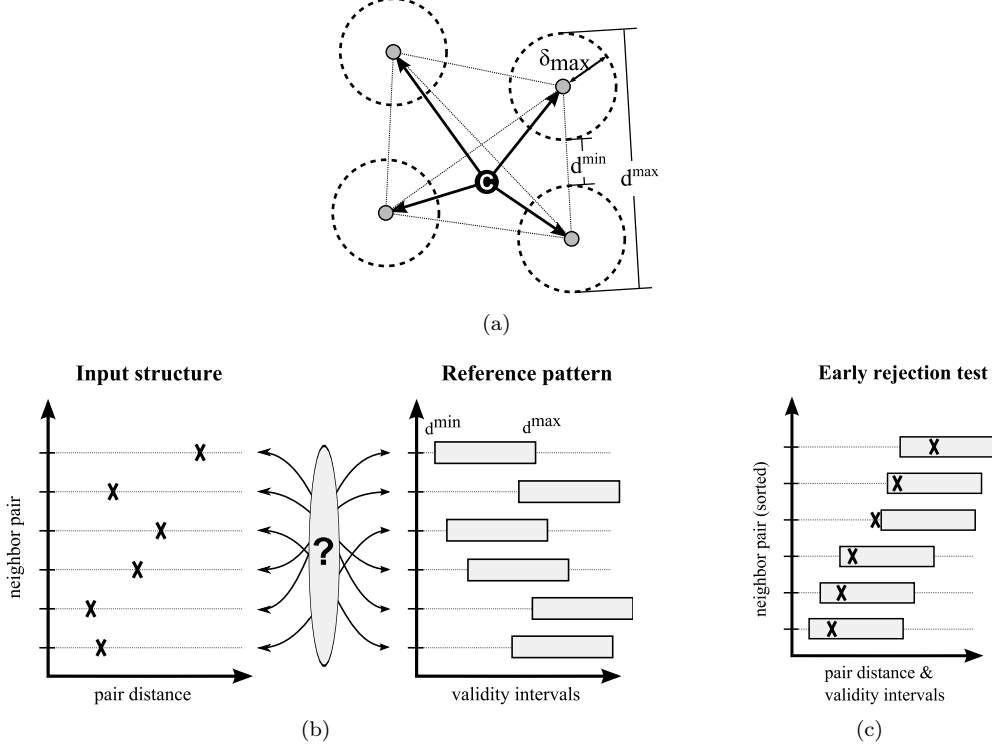


Figure 1. (a) Schematic picture of a low-symmetry coordination structure around a central atom. Dashed circles indicate the maximum distance a neighbor may deviate from its equilibrium position. This yields six min-max constraints on the mutual distances between the four neighbors. (b) For a positive match, a permutation of the neighbors must be found such that the actual distances fall into the intervals of the reference pattern. (c) By simply sorting the actual distances and the reference intervals, a quick rejection test can be performed without knowledge of the actual mapping.

To find a valid permutation map σ that fulfills condition 3 (or to confirm its non-existence), up to $N!$ possible permutations of the neighbors must be tested (figure 1(b)). To avoid an exhaustive search, the search space can, however, be reduced considerably by pruning the combinatorial search tree and employing a backtracking algorithm [18]. As an additional optimization step prior to the full combinatorial search, we perform an early rejection test on the entire coordination structure by sorting both the list of mutual distances $\{d_{ij}\}$ and the list of intervals $\{[d_{ij}^{\min}, d_{ij}^{\max}]\}$ in ascending order (figure 1(c)). If any of the distances falls outside the corresponding reference

interval, no valid permutation map can exist and the test structure does not match the coordination pattern.

The user needs to specify two control parameters for each NDA coordination pattern: The number of nearest neighbors to be taken into account (N) and the maximum admissible displacement (δ_{\max}). N must be at least three, should include complete shells, and, apart from that, be as small as possible for best efficiency. We also require that a coordination pattern includes enough neighbors such that each of them is itself a neighbor of at least two others in the same coordination pattern. This requirement will be needed to facilitate the consistent alignment of lattice orientations at each crystal site (see section 3). For the bcc lattice, for instance, this means that N must be 14; the second shell of neighbors must be included such that any two neighboring bcc atoms possess some common neighbors.

The maximum admissible displacement parameter δ_{\max} determines the tolerance of the identification process. In general one would want to use a large δ_{\max} to make the recognition of structures robust at high temperatures or in the presence of strong elastic distortions. On the other hand, a too large δ_{\max} parameter may lead to *false positives* when testing against multiple, only slightly different coordination patterns. That is, the local structure may match to multiple patterns if the agreement is within the specified tolerance for each of them. The resulting ambiguity will, in most cases, be resolved by the subsequent multi-atom analysis step (section 3), where we take the non-local atomic structure beyond the nearest neighbors into account as an additional criterion.

Note that we proposed the NDA only for identifying defective coordination structures that cannot be handled well with existing methods. In simple cases (such as perfect fcc, hcp, or bcc lattices), however, the conventional techniques such as the CNA might be the more economic choice. Thus, in practice, we use a combination of both the NDA and CNA to characterize local atomic structures. The CNA matching algorithm can be extended in a similar fashion such that it also yields the permutation map σ .

3. Multi-atom pattern analysis

Except for the simplest cases, most lattices consist of several atoms per unit cell, each having a different coordination structure. A local coordination analysis of individual atoms alone is not sufficient to unambiguously identify such extended structures. Let us take the 9R lattice [19] as an illustrative example. This close-packed structure (stacking ...ABCBCACAB...) is a repeating sequence of three atomic layers. One atomic layer has an fcc-like local coordination, while the other two contain atoms with an hcp-like arrangement of neighbors (see figure 8(a)). It is clear that the coordination structures of multiple atoms need to be taken into account simultaneously to identify such a lattice structure.

This is the task of the second algorithm described in the following sections. It searches for occurrences of a multi-atom pattern (derived from the lattice's primitive unit cell) in the input data. The search algorithm divides the crystal into contiguous domains that exhibit perfect long-range order. Note that many planar crystal defects such as stacking faults, surfaces, coherent crystal interfaces and grain boundaries have a structure that can be described in terms of a pattern that is repeated in two dimensions. Thus the same pattern matching algorithm can be used to find occurrences of such planar defects in a microstructure. Furthermore, small point-like

defects such as vacancy or interstitial clusters can also be identified and counted with the same universal algorithm.

3.1. Definition of a multi-atom pattern

A multi-atom pattern is a graph structure combining several coordination patterns into a single unit. Our representation scheme is based on so-called *periodic graphs* or *crystal nets*, which are used for topological descriptions of crystal structures in crystal chemistry [20]. Following the terminology given in [21], a multi-atom pattern is a node- and edge-labeled directed periodic connected graph $G_P = (V_P, E_P, \Sigma, p)$. Here, V_P is a set of nodes (atoms); E_P is a set of directed edges (bonds); Σ is a set of node labels (the catalog of coordination patterns), and p is a mapping function which assigns each node, $v_i \in V$, a coordination pattern $p(v_i) \in \Sigma$. A directed edge $(v_i, v_j, \mathbf{L}) \in E_P$ leads from node v_i to node v_j , and is labeled with a vector $\mathbf{L} \in \mathbb{R}^3$ that connects the two corresponding atoms in periodic Euclidean space.

A *path* is an alternating sequence of n nodes (v_1, v_2, \dots, v_n) and $n - 1$ consecutive edges $((v_1, v_2, \mathbf{L}_{12}), (v_2, v_3, \mathbf{L}_{23}) \dots (v_{n-1}, v_n, \mathbf{L}_{n-1,n}))$. A *circuit* is a closed path in which the first and the last node are the same. We use $\Delta \mathbf{L} = \sum_{i=1}^{n-1} \mathbf{L}_{i,i+1}$, to denote the sum of edge vectors in a path. The sum of edge vectors of a closed circuit is always an integer multiple of the repeat vectors $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3 \in \mathbb{R}^3$ of the unit cell, i.e. $\Delta \mathbf{L} = n_1 \mathbf{h}_1 + n_2 \mathbf{h}_2 + n_3 \mathbf{h}_3$ with $n_1, n_2, n_3 \in \mathbb{Z}$.

The outdegree $\deg(v_i)$ of a node v_i denotes the number of edges to neighboring nodes. According to section 2.2, the coordination pattern $p(v_i) \in \Sigma$, assigned to a node v_i , is essentially an ordered list $(\mathbf{R}_1, \dots, \mathbf{R}_N)$ of $N = \deg(v_i)$ vectors pointing from a center point to the neighbors. There is a one-to-one correspondence between the vectors of the coordination pattern and the incident edges of the node it is assigned to. However, by labeling each incident edge with an independent vector \mathbf{L} , which does not have to coincide with the corresponding vector \mathbf{R} of the coordination pattern, we add an extra level of indirection. This allows us to decouple the definition of coordination patterns from their usage in the multi-atom pattern. In particular, we can exploit rotational symmetries of structures and assign the same coordination pattern to multiple nodes which have a similar but rotated coordination structure.

A multi-atom pattern is either 3-periodic, 2-periodic or non-periodic, depending on whether it describes a lattice structure, a planar defect, or a point-like defect. For simple lattices such as fcc, the periodic graph consists of only one node that is connected to itself via a number of edges. The assigned edge vectors \mathbf{L} implicitly specify the geometry of the repeat unit (i.e. $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$).

In the case of ordered multi-component systems, each node is also labeled with a chemical species. This chemical information will serve as an additional criterion in the matching algorithm for cases where the structural information alone, given by the coordination patterns, is ambiguous (as in the L1₂ structure for instance).

The generation of a multi-atom pattern can be fully automated, and is based on the atomic coordinates of an ideal template structure provided by the user. The catalog of coordination patterns, Σ , is created by successively inspecting each atomic site in the template structure. If the local coordination structure does not match to one of the existing coordination patterns then a new pattern is created for that node and added to the catalog. Figure 2 depicts template cells for various structures from which multi-atom patterns and coordination patterns are derived. Here, only three distinct coordination patterns (indicated by the colors red, green, and blue) are

necessary to characterize the nodes in the multi-atom patterns.

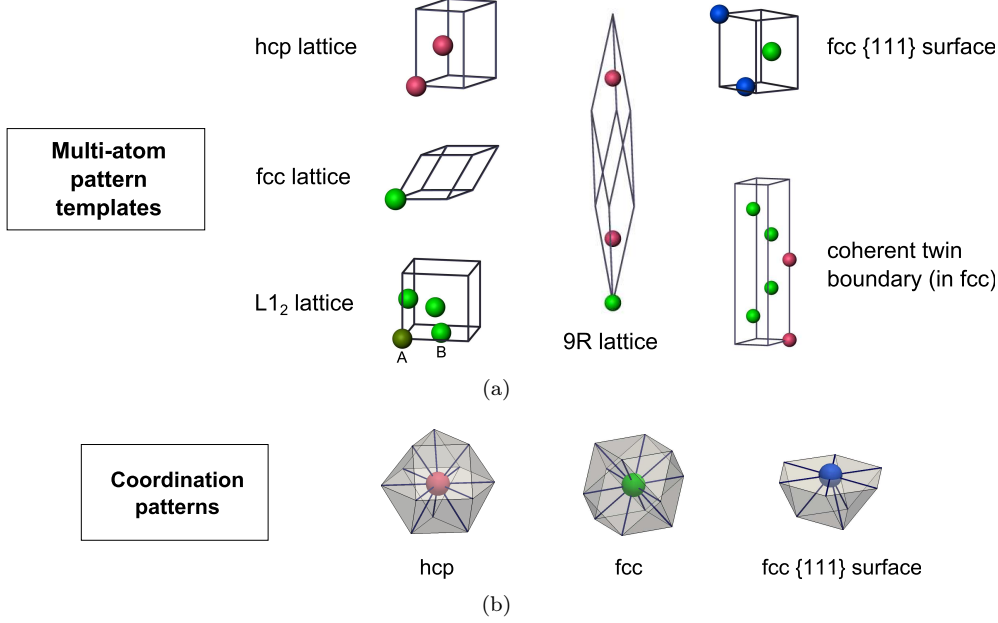


Figure 2. (a) Several template structures from which a catalog of multi-atom patterns and coordination patterns is derived. The first four structures (hcp, fcc, $L1_2$, and 9R) are 3-periodic lattices, while the other two structures (fcc surface and fcc twin boundary) are 2-periodic planar defects. (b) Three different coordination patterns occur in the six multi-atom patterns. The assignment of coordination patterns to the multi-atom pattern nodes is indicated by the colors red, green and blue.

3.2. Generation of the input graph

The pattern matching method that we use to find occurrences of a catalog of multi-atom patterns $\Pi = \{G_P\}$ in an atomistic simulation snapshot can be divided into two phases: First, the atomistic simulation snapshot is transformed into an input graph H_{in} by testing each atom against the patterns in the coordination pattern catalog Σ . In the second phase, we search for clusters of atoms (i.e. subgraphs of H_{in}) that topologically agree with one of the periodic pattern graphs $\{G_P\}$.

Let V_a be the set of atoms in the input snapshot, which constitute the nodes of the input graph H_{in} . We test each atom $a_c \in V_a$ against each coordination pattern $p \in \Sigma$ using an appropriate coordination pattern matching algorithm (CNA or NDA). A positive match yields a list $\sigma_p = (a_1, \dots, a_N) \subseteq V_a$ that maps the N bonds of p to an equal number of neighbors of a_c . The determination of the permutation map σ_p was discussed in section 2.4. We label a_c with the set of all positive matches, $\{\sigma_p\}$. Note that an atom may not match to any coordination pattern at all, then this set will be empty. Or it may match to multiple patterns, all being rather similar. Then $\{\sigma_p\}$ will contain multiple entries. Each match record σ_p implicitly defines a set of directed edges from atom a_c to neighboring atoms in H_{in} .

Given a match record σ_p , we can apply the coordination pattern's point symmetry

group S_p to it to generate other equivalent matches. We assume that an atom’s match list, $\{\sigma_p\}$, (implicitly) contains these equivalent matches too.

3.3. Pattern search algorithm

To search for occurrences of a pattern graph G_P in the input graph H_{in} we use a variation of Ullmann’s graph matching algorithm [22]. This algorithm starts by mapping a first node of G_P to a matching seed atom in H_{in} . Then neighbor nodes are successively mapped to matching neighbors of the seed atom to expand the partial match. The algorithm recursively continues with the neighbors’ neighbors until eventually the entire pattern graph G_P has been mapped to a subgraph of H_{in} . Unfruitful search paths are pruned, and if a pattern node cannot be associated with any available atom in H_{in} due to a violation of the isomorphism conditions, the algorithm backtracks and undoes the last assignment to continue with another branch of the state search tree.

For our application, Ullmann’s original algorithm has been modified in two aspects: The first modification pertains to the implicit representation of edges in the input graph H_{in} . In the previous section we pointed out that an atom can have several *potential* coordination structures (i.e. sets of edges to other atoms) given by the atom’s match list, $\{\sigma_p\}$. Multiple match records result from local point symmetries and the ambiguity of coordination patterns. Now we have to decide for one of these coordination structures. To this end we extend the search space of Ullmann’s graph matching algorithm to include each atom’s list of potential coordination structures.

A subgraph isomorphism found by the graph matching algorithm is then fully specified in terms of a set of tuples, $\mathcal{C} = \{(v_i, a_i, \sigma_p)\}$, that associate atoms of the input snapshot with nodes of the pattern graph. Here, $v_i \in V_P$ is a node of G_P , $a_i \in V_a$ is a corresponding atom in H_{in} , and σ_p is one of the atom’s match records, i.e. a map from the coordination pattern p to a set of neighbors of a_i (cf. section 3.2). A necessary isomorphism condition is that σ_p is a match record for the node’s coordination pattern $p(v_i)$, and the atom’s chemical species matches that of the pattern node.

The map σ_p can then be inverted such that, given a neighbor atom $a_j \in \sigma_p$ of a_i , we obtain its index in the coordination pattern $p(v_i)$. Together with the node v_i , this yields the corresponding edge $(v_i, v_j, \mathbf{L}) \in E_P$ in the multi-atom pattern. Thus, we have defined a lookup function $\mathbf{L}_C(a_i, a_j)$ that, given two neighboring atoms, yields a corresponding ideal bond vector in the unit cell. A second necessary isomorphism condition is $\mathbf{L}_C(a_i, a_j) = -\mathbf{L}_C(a_j, a_i)$ and $\mathbf{L}_C(a_i, a_j) + \mathbf{L}_C(a_j, a_k) + \mathbf{L}_C(a_k, a_i) = \mathbf{0}$ for every pair (a_i, a_j) and triplet (a_i, a_j, a_k) of neighbor atoms in the matched region of H_{in} . In other words, the lattice orientation picked at one site must match with those picked at neighboring sites. The graph matching algorithm uses these conditions to prune the combinatorial search tree.

Note that, since G_P is a periodic graph, it may match to a subgraph of H_{in} that is larger than G_P itself. Thus, the subgraph isomorphism \mathcal{C} is no longer bijective, and a single node $v_i \in V_P$ of G_P may be mapped to an arbitrary number of atoms $a_i \in V_a$ in H_{in} as long as perfect long-range order is not interrupted. For instance, let G_P be a lattice unit cell, then an entire crystallite is returned by a single invocation of the graph matching algorithm. We call the periodic subgraph isomorphism *complete* or a *cluster*, if it maps every pattern node $v_i \in V_P$ to at least one atom. A cluster \mathcal{C} is *maximal* if no more atoms at its perimeter can be added to it without violating the isomorphism conditions, i.e. without interrupting the long-range order. In the

following, $V(\mathcal{C}) \subseteq V_a$ denotes the set of crystal atoms that make up a cluster \mathcal{C} .

3.4. The vicinity criterion

In practice, the generation of maximal clusters can lead to counterintuitive results. Figure 3(a) depicts such a situation. Here, the periodic pattern G_P comprises four nodes labeled with the species A and B . The crystal being analyzed contains an anti-phase boundary (APB), which interrupts the long-range order. Since every other row matches to the periodic pattern, irrespective of the phase shift at the APB, the left cluster ‘bleeds’ into the right cluster.

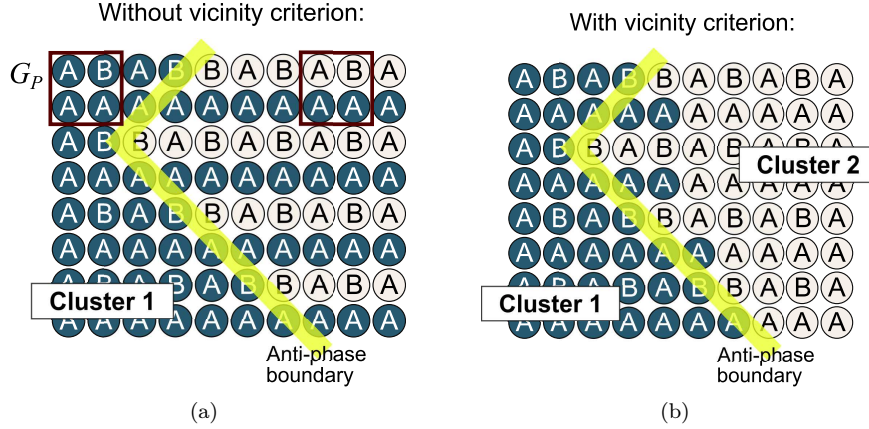


Figure 3. Both pictures show the same crystal as it is being decomposed into clusters with perfect long-range order. The biatomic crystal lattice contains an anti-phase boundary (APB). Without the vicinity criterion discussed in the text (left), the first cluster extends beyond the APB, since single rows of A atoms exhibit perfect order. This artifact can be avoided by enforcing a vicinity criterion (right), which ensures that an atom becomes part of a growing cluster only if all other atoms of the unit cell appear in its vicinity (and on the correct sites).

To mitigate such undesirable effects, one can add an additional isomorphism condition, which we call a *vicinity criterion*. An atom a may only become part of a cluster if all nodes of the pattern appear at least once in a neighborhood $n(a)$ of a . Note that we have some freedom in how we define this neighborhood.

The *distance* $d(a_i, a_j)$ between two atoms $a_i, a_j \in V(\mathcal{C})$ is the number of steps in the shortest path with initial node a_i and terminal node a_j that is completely contained in the cluster \mathcal{C} . The *neighborhood of size s* is the set $n_s(a) = \{a_j \in V(\mathcal{C}) : d(a, a_j) < s\}$. In our implementation we use the following vicinity criterion: Let $N_P = |V_P|$ denote the number of nodes in the periodic pattern graph. For a candidate atom a to become part of the cluster we require that the set of pattern nodes found in the neighborhood $n_s(a)$ contains at least s unique pattern nodes for any size $1 \leq s \leq N_P$. That is, with the s -th recursive step away from the candidate atom, we must have encountered at least s unique pattern nodes in its vicinity (in addition to the central node). If this criterion is not fulfilled then the candidate atom is rejected from the cluster (see figure 3(b)).

Note that alternative kinds of vicinity conditions are possible. The best choice depends, in general, on the size and the geometry of the structural unit cell.

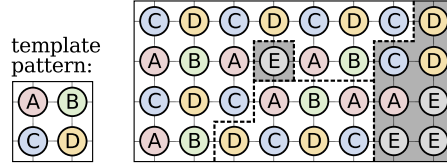


Figure 4. Illustration of the decomposition of a crystal into clusters. The letters *A-E* represent the local coordination structure or species of an atom. The clustering algorithm finds two periodic clusters (subgraphs) in the test structure that match exactly the pattern template. Two other regions remain (shaded atoms), which have an unidentified structure. Note that the *D* atom in the upper right corner is not part of a cluster even though it is located on a correct lattice site. It has been rejected because it violates the vicinity criterion by being more than two neighbor distances away from the next *A* or *B* atom.

3.5. Crystal defects

The input graph H_{in} is scanned for one multi-atom pattern from the catalog $\Pi = \{G_P\}$ at a time. Once a cluster has been found by the pattern search algorithm, all atoms belonging to that cluster are marked and will be ignored in subsequent searches. Thus, any atom can only become part of at most one cluster. Figure 4 schematically shows the outcome of the clustering algorithm for a pattern consisting of four nodes.

So far we have focused on the identification of periodic crystal lattices. Let us now consider the application of the pattern matching algorithm to lattice defects. In contrast to crystal lattices, defects cannot exist on their own: They are always embedded in a specific lattice, or, in the case of interfaces, located in between two crystallite clusters. It makes sense to include the specification of the surrounding lattice type (or types) in the description of a crystal defect pattern to avoid ambiguities. Let us take intrinsic stacking faults (ISF) in the fcc lattice as an example (see figure 5). Such defects are constituted by two atomic layers having the same coordination structure as atoms in the hcp lattice (stacking sequence ...ABC|AC|ABC...). Thus, one could regard ISFs as a thin hcp-phase layer that is separated from the fcc crystal by two coherent interfaces. Or conversely, one could consider a bulk hcp crystal as a large stacking of ISF defects. Neither would be completely correct. We can resolve this ambiguity by requiring ISF defects to be always bordered by fcc clusters on both sides and by giving defect clusters precedence over lattice clusters.

To this end we split the multi-atom pattern catalog into two partial catalogs, Π_{lat} and Π_{def} , that contain only lattice patterns and defect patterns respectively. Two separate search passes are performed, one for the set Π_{lat} and one for Π_{def} . Clusters found during any one pass may not overlap with each other, but defect clusters found during the second pass can overlap with lattice clusters found during the first pass. To prevent bulk hcp atoms (in a genuine hcp phase) from being labeled as ISF defects, we extend the multi-atom pattern for ISFs to include four nodes as shown in figure 5: the two hcp-coordinated atoms forming the defect core, and two additional fcc atoms on either side of the defect plane. These fcc nodes at the perimeter of the defect core overlap with the surrounding lattice and function as a filter. Thereby is ensured that the ISF pattern can be matched only to atomic arrangements embedded in a fcc crystal and not in other crystal phases.

In a defect multi-atom pattern, the lattice-like nodes at the perimeter of the defect

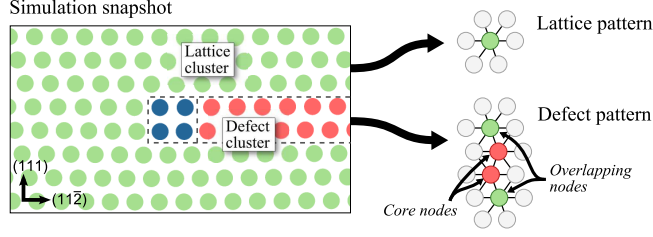


Figure 5. Cross-sectional view of a defective fcc crystal that contains a stacking fault bounded by a partial dislocation. The crystal has been decomposed into two clusters (and the unidentified dislocation core region). The multi-atom pattern for the stacking fault defect comprises additional overlap nodes to ensure that occurrences of the defect are always embedded in an fcc lattice.

core are flagged as *overlap nodes* meaning that, even though they must be present in the input structure, they are not considered part of the defect itself. That is, the corresponding atoms will not be incorporated into the defect cluster, but stay part of the surrounding lattice clusters.

4. Cluster analysis

In section 3.3 we have introduced the lookup function $\mathbf{L} = \mathbf{L}_{\mathcal{C}}(a_i, a_j)$, which, given a pair of neighbor atoms in a cluster \mathcal{C} , returns the ideal vector $\mathbf{L} \in \mathbb{R}^3$ connecting the two corresponding nodes in the multi-atom pattern. \mathbf{L} is a *cluster-space* vector; it is given in the reference frame of the structure template cell from which the multi-atom pattern was derived. Correspondingly, $\mathbf{r} = \mathbf{r}(a_i, a_j) = \mathbf{r}_j - \mathbf{r}_i$ is the vector connecting the two atoms in the simulation's frame of reference. Accordingly, \mathbf{r} is a *spatial* vector.

Note that, in the spatial configuration, a cluster can be elastically distorted (and with it the interatomic vectors $\mathbf{r}(a_i, a_j)$). In the cluster's reference frame, in contrast, atoms are positioned on ideal sites at all times. That is, the vectors $\mathbf{L}_{\mathcal{C}}(a_i, a_j)$ are unaffected by thermal or elastic perturbations of the atomic positions.

4.1. Cluster orientation

All atoms of a cluster \mathcal{C} form a contiguous periodic pattern with long-range order, that has an arbitrary orientation with respect to the simulation coordinate system. We can compute an affine transformation matrix $\mathbf{F}_{\mathcal{C}}$ that connects the cluster's frame with the spatial frame in a least-square sense:

$$\begin{aligned} \mathbf{F}_{\mathcal{C}} &= \mathbf{V}^{-1} \mathbf{W} \\ \mathbf{V} &= \sum_{(a_i, a_j)} \mathbf{L}_{\mathcal{C}}(a_i, a_j) \otimes \mathbf{L}_{\mathcal{C}}(a_i, a_j) \\ \mathbf{W} &= \sum_{(a_i, a_j)} \mathbf{L}_{\mathcal{C}}(a_i, a_j) \otimes \mathbf{r}(a_i, a_j) \end{aligned} \tag{4}$$

Here, the summations are over all neighbor pairs (a_i, a_j) in the cluster; \otimes denotes the outer product of two vectors. $\mathbf{F}_{\mathcal{C}}$ characterizes the average, macroscopic orientation of the crystallite cluster, including rigid-body rotations and elastic stretches. The lattice

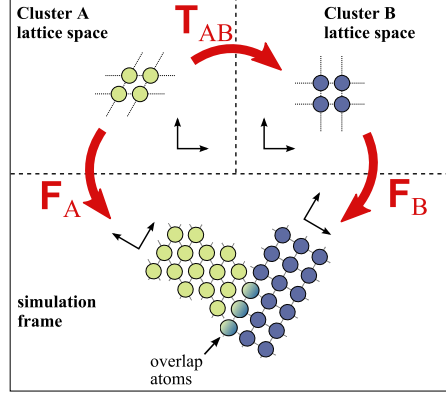


Figure 6. Illustration of the different reference frames used in the description of cluster orientations and cluster transitions.

orientation might vary locally though due to elastic distortions of the crystal. Thus, in general, the macroscopic orientation agrees only approximately with the microscopic orientation, i.e. $\mathbf{F}_C \mathbf{L}_C(a_i, a_j) \approx \mathbf{r}(a_i, a_j)$.

4.2. Cluster transition matrices

Let \mathcal{A} and \mathcal{B} be two clusters. The transformation matrix $\mathbf{T}_{\mathcal{AB}}$, relating their cluster-space vectors (i.e. $\mathbf{L}_{\mathcal{B}} = \mathbf{T}_{\mathcal{AB}} \mathbf{L}_{\mathcal{A}}$), is given approximately by

$$\mathbf{T}_{\mathcal{AB}} \approx \mathbf{F}_{\mathcal{B}}^{-1} \mathbf{F}_{\mathcal{A}}. \quad (5)$$

However, if \mathcal{A} and \mathcal{B} are two directly adjacent clusters (for instance, a crystal cluster and an embedded stacking fault defect as shown in figure 5), we may compute $\mathbf{T}_{\mathcal{AB}}$ exactly by making use of the overlap nodes introduced in section 3.5. Let a_o be an atom at the perimeter of the crystal defect that has been mapped to both a node of the lattice cluster \mathcal{A} and an overlap node of the defect cluster \mathcal{B} . Thus, each neighbor vector $\mathbf{r}(a_o, a_j)$ of this atom is simultaneously mapped to both reference frames, i.e. to the ideal vectors $\mathbf{L}_{\mathcal{A}}(a_o, a_j)$ and $\mathbf{L}_{\mathcal{B}}(a_o, a_j)$. This enables us to determine the cluster transition matrix $\mathbf{T}_{\mathcal{AB}}$ exactly:

$$\begin{aligned} \mathbf{T}_{\mathcal{AB}} &= \mathbf{V}^{-1} \mathbf{W} \\ \mathbf{V} &= \sum_{a_j} \mathbf{L}_{\mathcal{A}}(a_o, a_j) \otimes \mathbf{L}_{\mathcal{A}}(a_o, a_j) \\ \mathbf{W} &= \sum_{a_j} \mathbf{L}_{\mathcal{A}}(a_o, a_j) \otimes \mathbf{L}_{\mathcal{B}}(a_o, a_j) \end{aligned} \quad (6)$$

Here, the summation is over all neighbors a_j of the overlap atom a_o . Note that $\mathbf{T}_{\mathcal{AB}}$ does not depend on which overlap atom we pick for its calculation since cluster-space vectors are not affected by local elastic distortions.

Figure 6 schematically depicts the transitions between the cluster frames and the spatial frame.

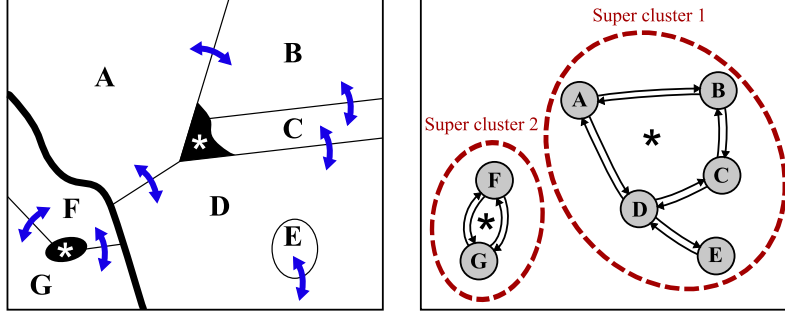


Figure 7. Left: schematic picture of a microstructure, which has been decomposed into clusters (letters A-G). Solid black regions do not belong to a cluster and have an unidentified structure. Asterisks (*) indicate possible locations of disclinations. Right: Abstract representation of the cluster network as a graph. Each cluster possesses a frame of reference that is tied to the local lattice orientation. An edge connecting two nodes represents a transformation matrix that relates the frame of reference of a cluster to that of an adjacent cluster.

4.3. The cluster graph

A *cluster graph* is an edge-labeled symmetric directed graph, $G_C = (V, E)$, generated from the described cluster adjacency analysis. Here, $V = \{\mathcal{C}_i\}$ is the set of clusters found in a simulation snapshot; E is the set of directed edges (transitions) connecting two neighboring clusters in the graph. A directed edge $(\mathcal{C}_i, \mathcal{C}_j, \mathbf{T}_{ij}) \in E$ leads from cluster \mathcal{C}_i to cluster \mathcal{C}_j , and is labeled with a transition matrix that transforms cluster-space vectors from cluster \mathcal{C}_i to cluster \mathcal{C}_j . The calculation of \mathbf{T}_{ij} from the overlapping region of \mathcal{C}_i and \mathcal{C}_j has been discussed in the preceding section. For every edge $(\mathcal{C}_i, \mathcal{C}_j, \mathbf{T}_{ij})$ there exists a reverse edge $(\mathcal{C}_j, \mathcal{C}_i, \mathbf{T}_{ij}^{-1})$. Clusters which are spatially separated, or which are separated by an unidentified type of crystal interface are not connected by an edge in the cluster graph.

Figure 7 schematically shows a cluster graph generated from a crystalline microstructure that was decomposed by the pattern matching algorithm.

4.4. Super cluster sets

The described graph representation of clusters and their transitions allows us to determine the transition matrices for pairs of non-adjacent clusters. Consider, for example, a twin boundary which separates two crystallites. The pattern matching algorithm decomposes such a bicrystal into two lattice clusters, \mathcal{C}_1 and \mathcal{C}_2 , and a defect cluster, \mathcal{C}_{TB} , for the core region of the twin boundary. The defect cluster overlaps with both lattice clusters, allowing us to compute the two transition matrices $\mathbf{T}_{\text{TB},1}$ and $\mathbf{T}_{\text{TB},2}$ according to equation 6. The transition matrix $\mathbf{T}_{12} = \mathbf{T}_{\text{TB},2}(\mathbf{T}_{\text{TB},1})^{-1}$ then describes the relative orientation of the two crystallites on either side of the grain boundary. Note that \mathbf{T}_{12} is the *ideal* crystallographic misorientation associated with the perfect grain boundary, which is unaffected by any superimposed elastic strains. The macroscopic misorientation of the two crystallites *including* elastic deformations and the effect of secondary grain boundary dislocations is given by equation 5.

The set $S = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_{\text{TB}}\}$ forms a so-called *super cluster*, since we can obtain a transition matrix for any pair of clusters in the set. That is, a cluster-space vector \mathbf{L} ,

specified in the reference frame of one cluster, can be transformed to any other cluster in the set. Formally, a *super cluster* $S \subseteq V$ is a *connected component* of the cluster graph defined in section 4.3 (see figure 7). The transition matrix \mathbf{T}_{ij} between two clusters $\mathcal{C}_i, \mathcal{C}_j \in S$ can be determined by finding a path connecting the initial node \mathcal{C}_i and the terminal node \mathcal{C}_j in the cluster graph. The transition matrix is computed by concatenating the consecutive transition matrices in the path.

It might be that the resulting matrix is path-dependent, i.e., it depends on how we traverse the cluster graph from \mathcal{C}_i to \mathcal{C}_j . Given two distinct paths from \mathcal{C}_i to \mathcal{C}_j , which yield the transition matrices $\mathbf{T}_{ij}^{(1)}$ and $\mathbf{T}_{ij}^{(2)}$ respectively, we can join them to form a closed circuit. We call such a circuit a Frank-Nabarro circuit[23], which is associated with a net transformation $\mathbf{M} = \left(\mathbf{T}_{ij}^{(2)}\right)^{-1} \mathbf{T}_{ij}^{(1)}$. The circuit encloses a disclination if $\mathbf{M} \neq \mathbf{I}$. The rotation matrix \mathbf{M} measures the rotational mismatch of the disclination defect.

5. Examples

5.1. Structural analysis of precipitates

We have applied the described multi-atomic pattern recognition algorithm to a simulation of a Fe-Cu multi-phase alloy. In this simulation study, a combination of Metropolis Monte Carlo sampling (variance-constrained semi-grandcanonical ensemble [24]) and conventional MD time integration has been used to determine the equilibrium structure of Cu-rich precipitates in the Fe-rich bcc matrix. Monte Carlo transmutation steps were used to find the equilibrium distribution of Cu atoms at a prescribed temperature, while alternating MD steps allow the positional degrees of freedom to relax simultaneously. This makes it possible for structural phase transformations to occur within the simulation. Starting off from a random distribution of Cu atoms in the Fe matrix, the Cu atoms precipitate to form a spherical particle. Under certain temperature and concentration conditions, the structure of the cluster changes from bcc to a multiply-twinned 9R structure (herringbone structure) [19] as shown in figure 8(a).

The conventional common neighbor analysis is able to identify the local coordination structure of individual atoms in the matrix (bcc) and inside the precipitate (alternating fcc and hcp layers). But it fails to classify atoms in the transition region where the lattice constant changes gradually from bcc-Fe to bcc-Cu to 9R-Cu, since the CNA is restricted to a fixed neighbor cutoff radius. The newly introduced neighbor distance analysis does not suffer from this limitation, as it acquires the right number of neighbors adaptively for each atom.

When fed with an adequate catalog of template structures, the multi-atom pattern analysis algorithm is able to identify the 9R phase as well as twin boundaries and stacking faults in the 9R lattice correctly (figure 8(b)). Note that there are several different types of stacking faults in the 9R phase.

The fully-automated pattern analysis can effectively determine almost every atom's role in the crystal, and classify the type of lattice or defect it is part of. Such information could be used, for instance, to quantify the density of various defect types as function of the precipitation conditions.

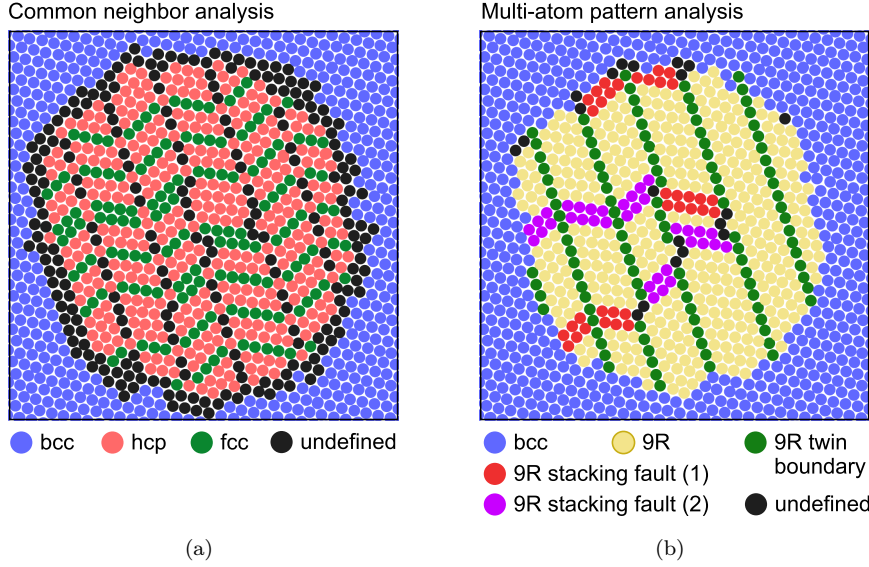


Figure 8. Cross-sectional view of a Cu-rich precipitate in bcc-Fe. Atom colors indicate local structure type as determined by (a) conventional common neighbor analysis and (b) multi-atom pattern analysis.

5.2. Point defect identification

The pattern matching method described in this paper can serve as a simple tool for counting and tracking point defects in complex atomistic simulations where classical analysis methods fail. Figure 9 displays a processed snapshot of a molecular dynamics study [25] of hydrogen embrittlement of iron under extreme conditions. The nanocrystalline Fe sample contains high densities of grain boundaries, dislocations, vacancies, and hydrogen atoms. The material is uniaxially strained to study the interaction of gliding dislocations with vacancies, hydrogen interstitials, and vacancy-hydrogen clusters.

Conventional identification methods for vacancies and interstitials [26] such as the Wigner-Seitz (or Voronoi cell) method cannot be used in the presence of dislocations and grain boundaries as they require the definition of a reference state of the crystal. Such a reference state would have to include all crystal defects except for the point defects, and is therefore hard to construct.

Our analysis method, in contrast, is based on a catalog of locally confined reference structures, which are rather easy to prepare. In the example shown in figure 9, we have employed the pattern matching technique to count the numbers of vacancies, di-vacancies, and tri-vacancies in the interior of the grains. In addition, the dislocation extraction algorithm (DXA) [4] has been used to identify dislocation lines in the bcc lattice.

6. Summary

A framework of computational methods has been devised to process the raw atomistic data (obtained from large-scale molecular dynamics simulations of crystalline

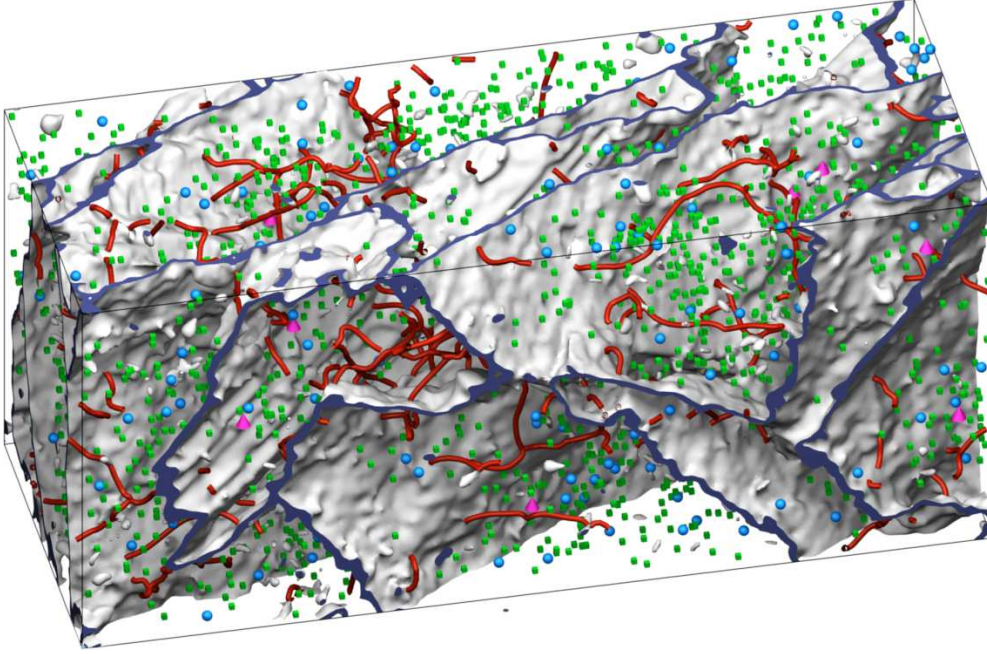


Figure 9. Visualization of the analysis data generated from a molecular dynamics simulation of dislocation-point defect interaction in Fe-H [25]. The multi-atom pattern analysis has been used to identify vacancies (green cubes), di-vacancies (blue spheres) and tri-vacancies (pink cones) in the Fe matrix. Hydrogen impurity atoms have been ignored during the analysis. The dislocation extraction algorithm (DXA) was used to extract the dislocation lines (red) and the geometric shapes of the grain boundaries (gray). The simulation cell volume is 51 x 20 x 22 nm, and the input data contained two million atoms.

materials) with the aim to generate an abstract description of the microstructure. Starting at the level of individual atoms, their arrangement is analyzed to identify larger structural units such as grains, interfaces, and other crystal defects.

The described processing sequence can be divided into three stages: First, the local arrangement of individual atoms is classified to provide the basis for the second analysis step, in which atoms are grouped into clusters based on the long-range pattern they form. In the third step, an abstract graph representation of the cluster network is generated, which includes information about structural phases, lattice orientations, and defect types. This data enables a wide spectrum of analyses at the microstructure level.

As an additional result, the atomistic data is enriched in the sense that the role of individual atoms as parts of larger structures is identified. This allows us to map atoms and their bond vectors to an ideal reference configuration. In forthcoming papers we will build upon this approach to develop new analyses. One application is the extension of the dislocation extraction algorithm [4] to partial dislocations and grain boundary dislocations. Since determining the Burgers vector of such dislocations requires Burgers circuits that pass through stacking faults or grain boundaries, we can make use of the described mapping of such defects to an ideal reference configuration. Another application of the described methods, which we present elsewhere, is the

decomposition of the atomic-level strain field into elastic and plastic parts.

Acknowledgments

The author thanks Paul Erhart for helpful discussions. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

- [1] J. D. Honeycutt and H. C. Andersen. Molecular dynamics study of melting and freezing of small Lennard-Jones clusters. *J. Phys. Chem.*, 91(19):4950–4963, 1987.
- [2] C. L. Kelchner, S. J. Plimpton, and J. C. Hamilton. Dislocation nucleation and defect structure during surface indentation. *Phys. Rev. B*, 58(17):11085, 1998.
- [3] A. Stukowski. Visualization and analysis of atomistic simulation data with OVITO – the Open Visualization Tool. *Modelling Simul. Mater. Sci. Eng.*, 18:015012, 2010. Software available at <http://ovito.org/>.
- [4] A. Stukowski and K. Albe. Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. *Modelling Simul. Mater. Sci. Eng.*, 18(8):085001, 2010.
- [5] A. Stukowski and K. Albe. Dislocation detection algorithm for atomistic simulations. *Modelling Simul. Mater. Sci. Eng.*, 18:025016, 2010.
- [6] A. S. Keys, C. R. Iacovella, and S. C. Glotzer. Characterizing complex particle morphologies through shape matching: Descriptors, applications, and algorithms. *Journal of Computational Physics*, 230(17):6438–6463, 2011.
- [7] Ju Li. Atomeye: an efficient atomistic configuration viewer. *Model. Simul. Mater. Sci. Eng.*, 11(2):173–177, 2003. <http://mt.seas.upenn.edu/Archive/Graphics/A/>.
- [8] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comp. Phys.*, 117(1):1, 1995. Software available at <http://lammps.sandia.gov/>.
- [9] H. Tsuzuki, P. S. Branicio, and J. P. Rino. Structural characterization of deformed crystals by analysis of common atomic neighborhood. *Comput. Phys. Commun.*, 177(6):518–523, 2007.
- [10] N. Lümmer and T. Kraska. Common neighbour analysis for binary atomic systems. *Model. Simul. Mater. Sci. Eng.*, 15(3):319–334, 2007.
- [11] G. J. Ackland and A. P. Jones. Applications of local crystal structure measures in experiment and simulation. *Phys. Rev. B*, 73(5):054104, 2006.

- [12] J. L. Finney. Random packings and structure of simple liquids. 1. geometry of random close packing. *Proceedings of the Royal Society of London A.*, 319:479, 1970.
- [13] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley, Chichester, 2nd edition, 2000.
- [14] D. Faken and H. Jonsson. Systematic analysis of local atomic structure combined with 3d computer graphics. *Comput. Mater. Sci.*, 2(2):279–286, 1994.
- [15] C. S. Hsu and A. Rahman. Interaction potentials and their effect on crystal nucleation and symmetry. *The Journal of Chemical Physics*, 71(12):4974–4986, 1979.
- [16] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, 1975.
- [17] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3:209–226, 1977.
- [18] D. E. Knuth. *The Art of Computer Programming*, volume 4A: Combinatorial Algorithms. Upper Saddle River, Addison-Wesley Professional, 2011.
- [19] F. Ernst, M. W. Finnis, D. Hofmann, T. Muschik, U. Schönberger, U. Wolf, and M. Methfessel. Theoretical prediction and direct observation of the 9R structure in Ag. *Phys. Rev. Lett.*, 69(4):620–623, 1992.
- [20] O. Delgado-Friedrichs and M. O’Keeffe. Identification of and symmetry computation for crystal nets. *Acta Crystallographica Section A*, 59(4):351–360, Jul 2003.
- [21] O. Delgado-Friedrichs and M. O’Keeffe. Crystal nets as graphs: Terminology and definitions. *Journal of Solid State Chemistry*, 178(8):2480–2485, 2005.
- [22] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23:31–42, January 1976.
- [23] F. R. N. Nabarro. *Theory of Crystal Dislocations*. Clarendon, Oxford, 1967.
- [24] B. Sadigh, P. Erhart, A. Stukowski, A. Caro, E. Martinez, and L. Zepeda-Ruiz. A scalable parallel Monte Carlo algorithm for atomistic simulations of precipitation in alloys. *arXiv:1012.5082v1*, 2010.
- [25] Ju Li and Suzhi Li. Private communication. June 2011.
- [26] K. Nordlund, M. Ghaly, R. S. Averback, M. Caturla, T. Diaz de la Rubia, and J. Tarus. Defect production in collision cascades in elemental semiconductors and fcc metals. *Phys. Rev. B*, 57(13):7556–7570, 1998.